



---

Research Institute for Advanced Computer Science  
NASA Ames Research Center

**"A Preliminary Data Model for Orbital Flight Dynamics in  
Shuttle Mission Control"**

**Authors: John O'Neill and Valerie L. Shalin**

**RIACS Technical Report 00.09  
Sept 2000**

# **"A Preliminary Data Model for Orbital Flight Dynamics in Shuttle Mission Control"**

**Authors: John O'Neill and Valerie L. Shalin**

The Research Institute for Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044 (410) 730-2656

---

Work reported herein was supported by NASA via Cooperative Agreement NCC 2-1006 between NASA and the Universities Space Research Association (USRA). Work performed at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035-1000

## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>3</b>
<b>2</b>	<b>An Information-Centric Approach.....</b>	<b>3</b>
<b>3</b>	<b>The Orbital Flight Dynamics Domain .....</b>	<b>4</b>
<b>4</b>	<b>A Data Model for Orbital Flight Dynamics .....</b>	<b>7</b>
4.1	Context Diagram .....	7
4.2	Data Model.....	9
<b>5</b>	<b>Mapping the Data Model to Orbital Flight Dynamics Current Information Systems.....</b>	<b>10</b>
5.1	Semantic Analysis.....	11
5.1.1	Managing Changing Information.....	11
5.2	Multiple Sets of Data Stores .....	12
5.3	Database issues .....	13
5.4	Constraints on Work Practice - Rendezvous Planning.....	14
5.4.1	Discussion.....	16
<b>6</b>	<b>Envisioning the Future Work of Orbital Flight Dynamics.....</b>	<b>16</b>
<b>7</b>	<b>Conclusions .....</b>	<b>19</b>
<b>8</b>	<b>References.....</b>	<b>20</b>

# 1 Introduction

The Orbital Flight Dynamics group in Shuttle Mission Control is investigating new user interfaces in a project called RIOTS [RIOTS 2000]. Traditionally, the individual functions of hardware and software guide the design of displays, which results in an aggregated, if not integrated interface. The human work system has then been designed and trained to navigate, operate and integrate the processors and displays. The aim of RIOTS is to reduce the cognitive demands of the flight controllers by redesigning the user interface to support the work of the flight controller.

This document supports the RIOTS project by defining a preliminary data model for Orbital Flight Dynamics. Section 2 defines an information-centric perspective. An information-centric approach aims to reduce the cognitive workload of the flight controllers by reducing the need for manual integration of information across processors and displays. Section 3 describes the Orbital Flight Dynamics domain. Section 4 defines the preliminary data model for Orbital Flight Dynamics. Section 5 examines the implications of mapping the data model to Orbital Flight Dynamics current information systems. Two recurring patterns are identified in the Orbital Flight Dynamics work: the iteration/rework cycle and the decision-making/information integration/mirroring role relationship. Section 6 identifies new requirements on Orbital Flight Dynamics work and makes recommendations based on changing the information environment, changing the implementation of the data model, and changing the two recurring patterns.

## 2 An Information-Centric Approach

Information-centric approaches focus on defining the complete information environment to support the work as shown in Figure 1. An information environment includes the information used in computer-based work systems by different applications and tools, and the information communicated between people over voice-loops and in face-to-face conversations. Information environments also include the information communicated in debriefings, conversations in offices and corridors, and the information that supports learning across sims, missions, and generations of members.

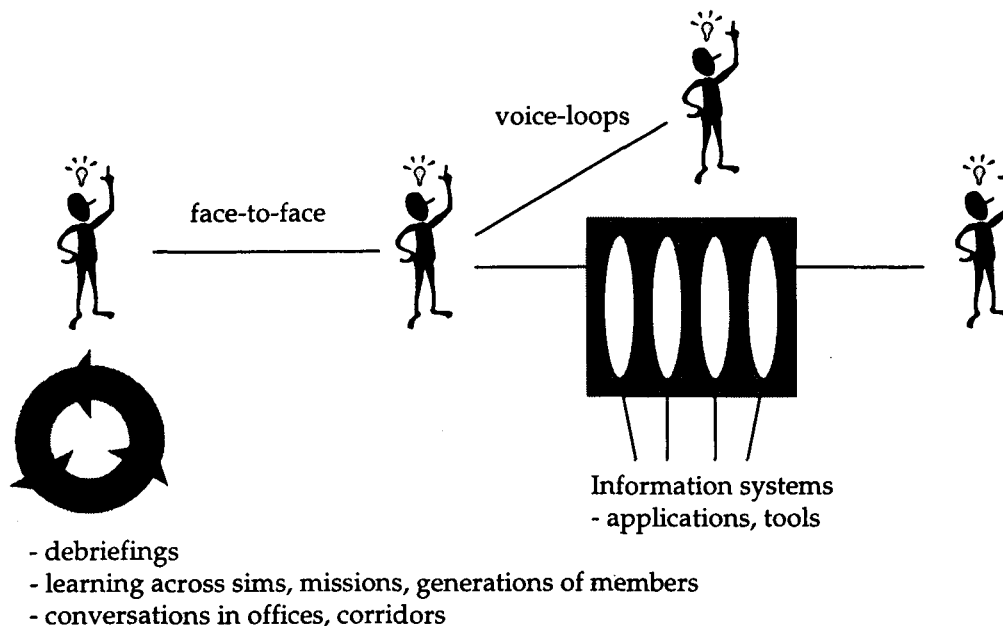


Figure 1. An information environment tailored for Mission Control

An information-centric approach for designing large-scale information systems to support people working focuses on leveraging and integrating information. In contrast, a functional, hardware and software driven approach to designing computer tools and applications results in unintegrated stove-piped monoliths. Criticism of this approach dates to at least DARPA's Pilot's Associate program in the late 1980's. Information-centric design approaches utilize layered architectures as shown in Figure 2. Layered architectures specifically decouple the user interface, the functionality, and the data storage to facilitate integration and reuse. Performance evidence supports the benefits of this decoupling and the resulting flexibility it permits (Shalin & Geddes, 1994; 1998).

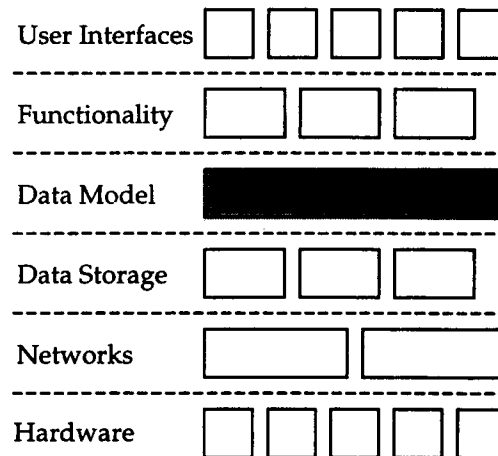


Figure 2. Layered Architectures

Information-centric design approaches are orthogonal to, but complementary with functional design approaches. Our focus is on designing and operating large-scale information systems that leverage and integrate the existing information infrastructure. The data model is the key to integrating data across applications, databases, networks, and computer systems. A *data model* is a logical construct that names things, defines the relationships between things, and describes who or what creates, updates, deletes, and touches the information.

Data models encourage the reuse of data across data stores. By utilizing a data model approach each application does not construct a database with all the data required to run the application. Instead, applications leverage and integrate existing data stores by using the vocabulary described in the data model.

Data models are not static constructs. Instead, data models are "living artifacts" that are continually updated as the work changes, as applications and tools are updated, and as new information system requirements emerge.

### 3 The Orbital Flight Dynamics Domain

Orbital Flight Dynamics is different to the other flight disciplines in Mission Control. Orbital Flight Dynamics work is more planning-oriented whereas other disciplines focus primarily on monitoring.

The central work of the Orbital Flight Dynamics group is managing the trajectories of objects in orbit in a correct and up-to-date manner. Objects being managed include the Orbiter (Space Shuttle), ISS, TDRS satellites, deployed objects, and debris.

Objects in orbit are represented or modeled in ephemerides. An *ephemeris* consists of a set of state vectors that describe the orbit at discrete points in time. At periodic intervals, a propagator (computational processor) uses a variety of physics models to generate the next state vector for

the object being tracked. An ephemeris is described by an anchor vector id (which identifies the initial state vector), the start time, the orbit number, the duration (in hours), and vehicle specific models.

Objects don't stay in the same orbit forever. The Orbital Flight Dynamics group is responsible for:

- planning maneuvers and integrating these maneuvers into the ephemeris
- incorporating weight changes and vents into the ephemeris
- changing the anchor vector id when more accurate tracking data is available
- changing the duration of the ephemeris (an artifact of earlier CPU limitations and tool limitations)
- conducting contingency planning, especially during ascent, descent, rendezvous deploy, and providing daily deorbit options for the Orbiter
- keeping the Orbiter onboard systems state vector up-to-date

Figure 3 shows the relationship between Orbital Flight Dynamics responsibilities. The properties of the ephemeris are used to compute a representation of the orbit of an object at discrete points in time and specify when planned changes will be made to the orbit. The properties of the ephemeris in combination with physical models are used by the propagators to calculate the next state vector. In contrast, the actual orbit of an object is continuous in nature and reflects influences not considered in the physical models. Consequently the actual orbit may diverge from the computed ephemeris, requiring periodic updates from the tracking data to modify the properties of the ephemeris.

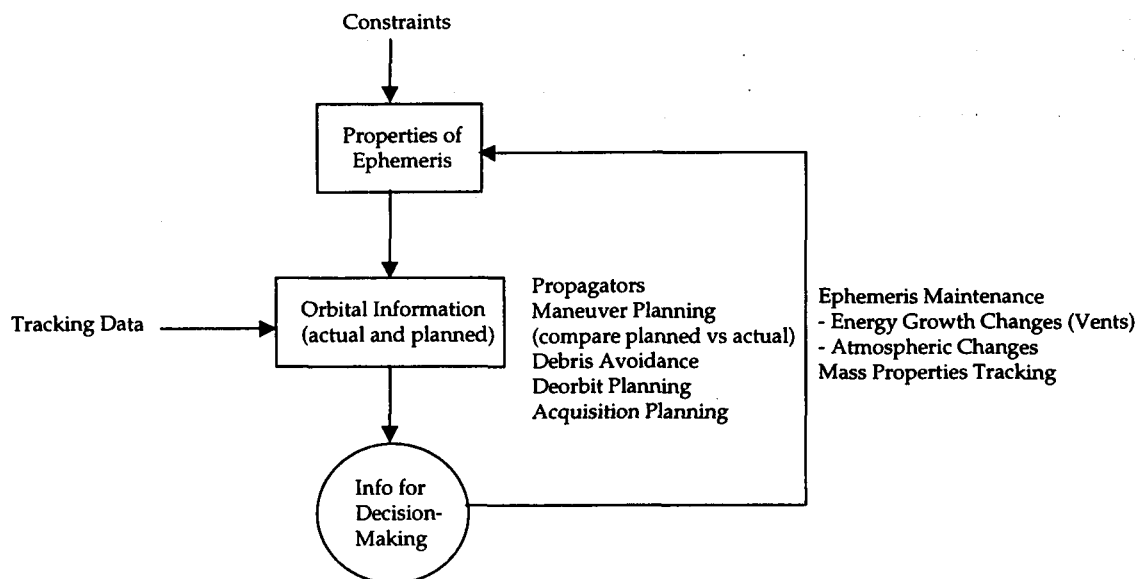


Figure 3. Orbital Flight Dynamics Work

The Orbital Flight Dynamics group currently uses 4 ephemerides on the MOC to model objects in orbit and a further 3 ephemerides specifically to track TDRS satellites. The relationship between these ephemerides and the Orbiter's actual orbit is shown in Figure 4 and is described below (in an idealized fashion):

- ephemeris 1 models the current orbit for the Orbiter and includes future maneuvers. The state vectors in ephemeris 1 may be periodically updated from tracking data to ensure the accuracy of the calculated orbit. Because a burn is never executed exactly as planned, after execution, the ephemeris will need to be updated with the actual  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  and TIG (time of

ignition). This is done by replacing the planned values already incorporated into the ephemeris with actual values, and ensures the accuracy of the orbit model

- ephemeris 2 and ephemeris 4 are used to conduct planning, and are initialized from a state vector in another ephemeris
- ephemeris 3 is used to model targets (e.g. ISS, deployed satellites, debris) and conduct rendezvous planning
- the TDRS ephemerides are used to track the TDRS satellites

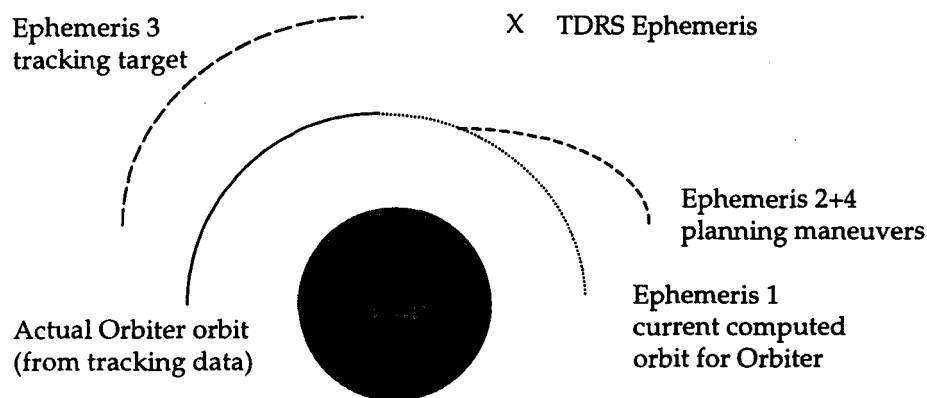


Figure 4. Ephemerides

One of the interesting features of the ephemerides is the relationships between them. For example, ephemeris 2 and ephemeris 4 may be related to ephemeris 1 through the use of a state vector for initialization. Ephemeris 3 and ephemeris 1 will be related for rendezvous maneuvers, and then share a common anchor vector id (state vector) for decoupling. One of the aims of the RIOTS project is to make explicit the relationships between ephemerides, reducing the cognitive load on the flight controllers associated with relying on their own memories. This has become increasingly urgent as the FDOs expand the number of ephemerides to accommodate a more complex environment, including a preponderance of planning intensive ground-up rendezvous flights, a growing debris field and mandated incorporation of a larger portion of the TDRS network per flight.

The roles in the Orbital Flight Dynamics group have responsibilities for making different types of decisions to ensure that the orbiter ephemeris is accurate and current :

- NAV is responsible for the accuracy of the state vectors and maintaining the vent timeline. State vectors are generated computationally and then compared to the tracking data coming from multiple sources including the Orbiter, ground stations, TDRS satellites, and GPS satellites. When the Orbiter is conducting a maneuver (via onboard data about thrust of accelerations) or is approaching another vehicle (within 40nm via the onboard Startracker sensors and radar), the Orbiter's onboard state vector is most accurate. One orbit after maneuvering the state vector derived from the tracking stations are most accurate. The vent timeline uses empirical data to model the effects of firing the thrusters to maintain the Orbiter's attitude.
- TRACK schedules ground sites and ensures the infrastructure is available to provide tracking data to NAV (the Track role will soon move to GC).
- TRAJ (Trajectory Officer) assists FDO in contingency planning, confirming burns, and keeping mass properties current.
- DYNAMICS is an expert on the internal workings of the MOC processors and is responsible for inputting FDO's verbal commands, examining the results for accurate input and

informing FDO when the requested changes are complete(see Section 5.3 for a rendezvous planning scenario) . Because of their role in manually updating the ephemerides, Dynamics typically takes responsibility for verbally informing the backroom when updates are in progress.

- PROFILE SUPPORT is responsible for shadowing rendezvous targeting problems on a separate workstation and conducting more extensive contingency planning (for example, looking at what-ifs, or if the rendezvous is aborted setting up for another attempt tomorrow).
- LANDING SUPPORT OFFICER is responsible for providing weather-related information for TRAJ to calculate deorbit information and rank landing sites. This information is continually updated during the ascent phase of Orbiter operations, and once per day throughout the rest of the mission.
- GA's (Group Administrator) are experts on the TSA tools and are responsible for applications in MCC and configuration information. Pre-mission, TSA is supported by the GA's. During a mission TSA is supported by on-call GA's with most of the troubleshooting being conducted by FDOs.
- TSU support in the future is still being decided, but will probably have a console position during missions similar to DYNAMICS.

A cursory analysis of these roles suggests that much of the work of NAV and Dynamics requires manual integration and confirmation of information across processors and displays. Section 5 will discuss the nature of the work in more detail in relation to the Orbital Flight Dynamics information environment. Section 6 will provide recommendations for work systems redesign.

## **4 A Data Model for Orbital Flight Dynamics**

This section defines the context for Orbital Flight Dynamics work and then defines a preliminary data model.

### **4.1 Context Diagram**

A context diagram defines the information flows into and out of some process. The context diagram in Figure 5 Orbital Flight Dynamics, focused exclusively on the creation and maintenance of the primary orbiter ephemeris. The purpose of a context diagram is to situate the work of Orbital Flight Dynamics in relationship to other work being conducted for Space Shuttle operations.



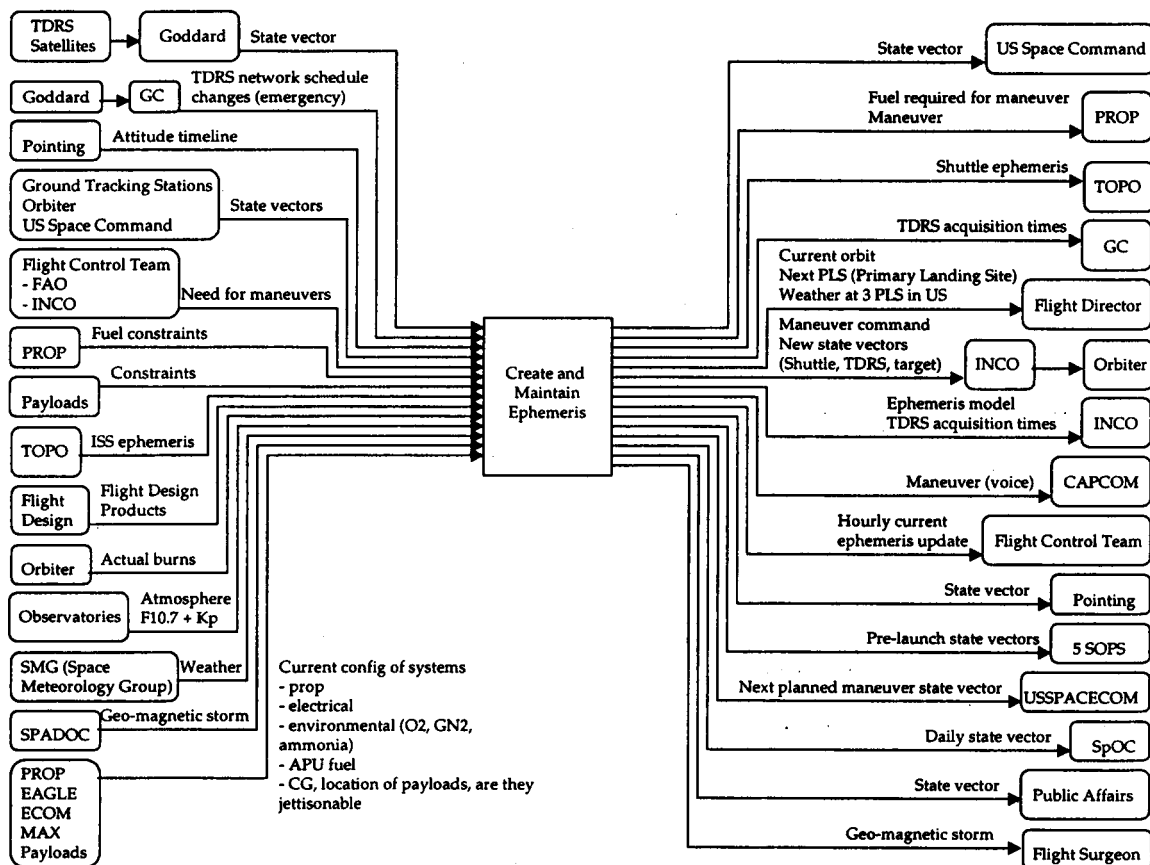


Figure 5. Context Diagram for Orbital Flight Dynamics

The left-hand side of the context diagram identifies sources of information. The sources may be roles within the Shuttle flight control team, or they may be external to mission control.

The line drawn from the source of information to the "Create and Maintain Ephemeris" box identifies a specific flow of information into Orbital Flight Dynamics. For example, "Pointing" supplies the "Attitude timeline".

The "Create and Maintain Ephemeris" box in the middle of the diagram represents the work of Orbital Flight Dynamics. Once the context diagram is defined, system design approaches normally focus on analyzing the center box, in this case "Create and Maintain Ephemeris" box, to recommend new work system designs.

The right-hand side of the context diagram represents information that flows from Orbital Flight Dynamics to customers. Most of the customers are other roles in the flight control team (and the INCOs are responsible for sending commands to the Orbiter). The line drawn from the "Create and Maintain Ephemeris" box to the customers identifies specific flows of information. For example, "TDRS acquisition times" are sent to the "GCs".

Some issues about the context diagram:

- Many of the information flows require manual intervention in the form of: calls over voice loops, sending faxes, receiving pieces of paper, manually moving information between systems.
- Differences in procedures, discipline-specific requirements for fidelity and sometimes lack of "trust" across flight disciplines sometimes result in duplication, rework and redundant effort leading to the possibility of errors from the use of out-of-date information. For example,

many of the state vector flows should actually be ephemeris flows. However, the state vectors are used by the flight discipline to create their own ephemeris using their own propagators e.g for ISS, both ISS MCC and Shuttle MCC maintain separate ISS ephemeris.

- Some information flows are a result of the FDOs being the point of contact for information from the outside world, and then passing on this information to the rest of the flight control team e.g. geo-magnetic storm information.

## 4.2 Data Model

A data model begins by naming things or objects in the domain, and then defining relationships between things from an information perspective [Chen, 1976]. A preliminary data model for Orbital Flight Dynamics is shown in Figure 6.

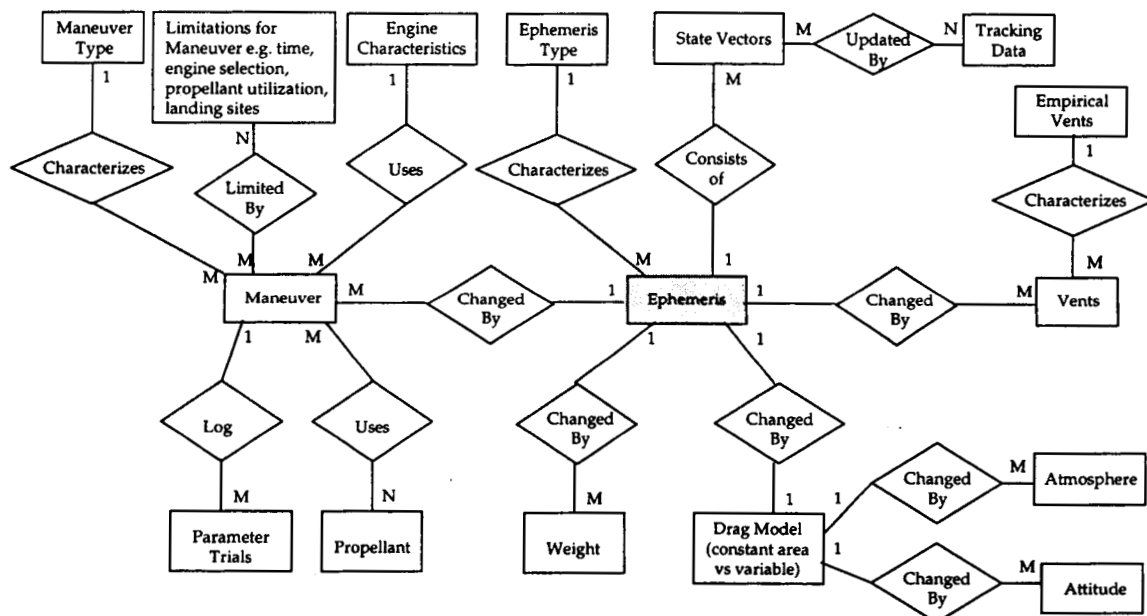


Figure 6. Data Model for Orbital Flight Dynamics

The starting point for reading a data model is the "Ephemeris" box in the middle of the figure. Each box represents an object or thing in the Orbital Flight Dynamics domain. Some examples of objects include maneuvers, state vectors and vents. The diamonds represent relationships between the objects.

A data model is read by starting in the middle of the figure and working outwards. For example, "an ephemeris consists of state vectors", "an ephemeris is changed by a maneuver", or "a maneuver is limited by constraints".

Each relationship is expressed using some cardinality (1, M, N). In each relationship, a "1" means that there is only one of that object, "M" means there be many of that object. For example, "an ephemeris may consist of multiple state vectors". The "N" means that the relationship may be multiple in both directions. For example, "each maneuver may be limited by multiple constraints" and "each constraint may limit multiple maneuvers".

Some features of the data model shown in Figure 5:

- the "Parameter Trials" object can log the parameters used for planning maneuvers on different iterations. By logging the parameter trials we would aim to develop heuristics for reducing the number of iterations by analyzing expert FDO performance, providing these

heuristics to the whole Orbital Flight Dynamics community, and then include an analysis of maneuver planning as part of the lessons learned after each mission/sim.

- the "Ephemeris Type" object supports the family resemblance issues between ephemerides
- the "Maneuver Type" object provides a place for storing decision aid advice for reducing parameter trials in developing a maneuver plan (see Section 5.3)

## 5 Mapping the Data Model to Orbital Flight Dynamics Current Information Systems

The second step in creating a data model is to determine the meta-data: who or what creates, updates, deletes, or touches the information. In the following section the existing data stores used by Orbital Flight Dynamics is mapped onto the preliminary data model as shown in Figure 7. The mapping sometimes relates existing tables to objects, sometimes to relationships. This is possibly due to the existing systems not being designed from an information-centric perspective.

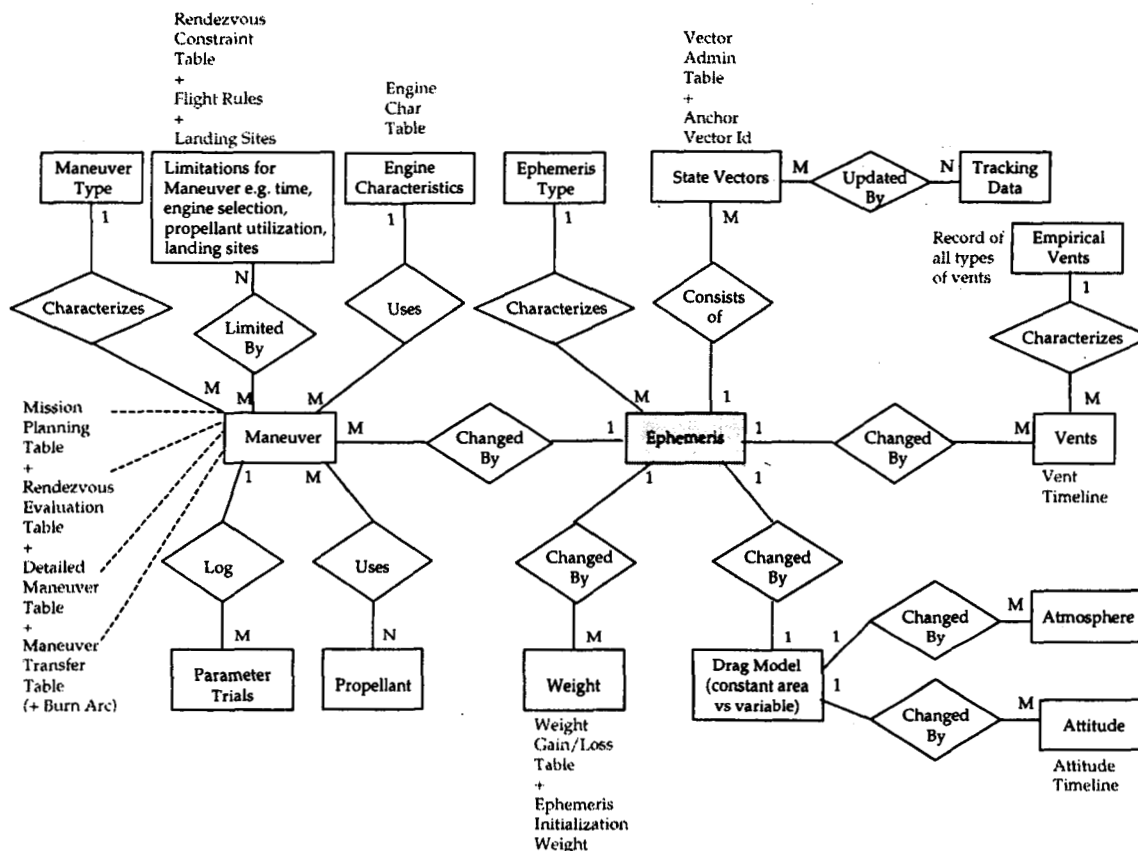


Figure 7. Mapping the Data Model to Existing Data Stores

The main tables being used for Orbital Flight Dynamics activities are mapped on to the data model in red writing, next to the object or relationship to which they conceptually relate. In this section we examine the semantic implications revealed by these mappings, the actual mappings for the different information systems (MOC, TSA, TSU), and the work practice limitations caused by the semantics.

## 5.1 Semantic Analysis

Examining the mapping of actual data stores to the data model as shown in Figure 6 reveals that the existing data model is only partially implemented in software. In particular:

- all the objects are implemented
- no relationships are implemented
- the "maneuver" object has a very complex implementation (see Section 5.3)

The objects are implemented purely as data sets, there is no implementation of semantics to identify the meaning of these data sets (the relationships in the data model represent semantic information). The lack of semantics adds to the flight controller's cognitive workload because the flight controller must keep track of the meaning and purpose of these data sets and the relationships between different things. For example:

- distinctions between ephemerides are implicit (inferred from weights, vectors, parameters and maneuvers). By convention in the TRAJ PROF STATUS screen (see Figure 7), ephemeris 1 is the current Orbiter ephemeris, ephemeris 2 is for Orbiter planning, ephemeris 3 is for the target, and ephemeris 4 is for planning. However, the purpose of two ephemerides can differ although all of the apparent indicators are the same (e.g., a TIG slip?). Similarly a discrepancy in number of maneuvers between two identical ephemerides can be the result of different reasoning, e.g., an additional step in the planning process, or a break-out plan. This leaves ample room for error for distinguishing between ephemerides during shift changes.
- there is no way of tracking the status of planning. For example, how does the next shift know whether maneuver planning is completed, whether the current parameters define a "good" maneuver, which ephemerides was used to conduct the maneuver planning and whether there has been divergence between the current ephemeris and planning ephemeris?
- there is no way of distinguishing state vectors in the vector administration table. While address "V39" may represent the current state vector, what do all the other entries represent and which ephemerides are using them?

### 5.1.1 Managing Changing Information

One attempt to enhance coordination within a distributed team is the TUP (Trajectory UPdate) field that indicates a change to the ephemeris. However, the TUP field adds cognitive demand. The TUP field is represented as a number. The flight controller must remember the current number for each of the extant ephemerides so that they know whether the number has changed (see Figure 8). A change in the number provides no indication of what has changed in the ephemeris or whether the change affects the trajectory in the near or long term (for example, a planned maneuver may have been inserted into the ephemeris that will be executed tomorrow).

000		TRAJ PROF STATUS		2310N	
EPHEMERIS 2 NSC 2 ARRAY NOT INITIALIZED					
PRIME ORBITER EPHEMERIS-EPH 1		PRIME TARGET EPHEMERIS-EPH 2		TUP INT 100 01 13 00 00	
EPH 1 PROFILE GOOD STATUS STAT		EPH 2 PROFILE GOOD STATUS STAT			
TUP NUMBER 1 KCON 1 0000		TUP NUMBER 1 KCON 1 0000			
NUMBER MOVES 0 EVAR 1 0000		NUMBER MOVES 0 EVAR 1 0000			
WTS INIT CURRENT AREA 1000 00		WTS INIT CURRENT AREA 1000 00			
VEH 210000 210000 0 INT OR OPT VAR		VEH 210000 210000 0 INT OR OPT VAR			
OMS 23752 0 23752 0 STDN OPT YES		OMS 23752 0 23752 0 STDN OPT NO*			
RCS 4592 0 4592 0		RCS 4592 0 4592 0			
EPHE VEH VT 210000 0		EPHE VEH VT 210000 0			
AVID ORB 101 ORB 101		AVID ORB 101 ORB 101			
GNTV 197 01 44 27 00 TORB 197 01 44 27 00		GNTV 197 01 44 27 00 TORB 197 01 44 27 00			
EPHL 48 00		EPHL 48 00			
EPHE 197 01 44 27 00 METB 000 00 00 00 00		EPHE 197 01 44 27 00 METB 000 00 00 00 00			
EPHE 199 01 44 27 00 METE 000 00 00 00 00		EPHE 201 01 44 27 00 METE 010 00 00 00 00			
EPH 3 PROFILE GOOD STATUS STAT		EPH 4 PROFILE GOOD STATUS STAT			
TUP NUMBER 1 KCON 1 0000		TUP NUMBER 1 KCON 1 0000			
NUMBER MOVES 0 EVAR 1 0000		NUMBER MOVES 0 EVAR 1 0000			
WTS INIT CURRENT AREA 1000 00		WTS INIT CURRENT AREA 1000 00			
VEH 210000 210000 0 INT OR OPT VAR		VEH 210000 210000 0 INT OR OPT VAR			
OMS 23752 0 23752 0 STDN OPT NOE		OMS 23752 0 23752 0 STDN OPT NOE			
RCS 4592 0 4592 0		RCS 4592 0 4592 0			
EPHE VEH VT 210000 0		EPHE VEH VT 210000 0			
AVID ORB 101 ORB 101		AVID ORB 101 ORB 101			
GNTV 197 01 44 27 00 TORB 197 01 44 27 00		GNTV 197 01 44 27 00 TORB 197 01 44 27 00			
EPHL 48 00		EPHL 48 00			
EPHE 197 01 44 27 00 METB 000 00 00 00 00		EPHE 197 01 44 27 00 METB 000 00 00 00 00			
EPHE 199 01 44 27 00 METE 000 00 00 00 00		EPHE 201 01 44 27 00 METE 010 00 00 00 00			
TORSS E PROFILE		TORSS U PROFILE		TORSS S PROFILE	
TUP NUMBER 1		TUP NUMBER 1		TUP NUMBER 1	
AVID TE 041		AVID TE 041		AVID TE 041	
GNTV 197 03 15 00 00		GNTV 197 03 15 00 00		GNTV 197 03 15 00 00	
EPHL 100		EPHL 100		EPHL 100	
EPHE 197 03 15 00 00		EPHE 197 03 15 00 00		EPHE 197 03 15 00 00	
EPHE 202 03 15 00 00		EPHE 202 03 15 00 00		EPHE 202 03 15 00 00	

TUP Field

Figure 8. TUP field for ephemeris on the Trajectory Profile Status display

Reducing the cognitive load on flight controllers requires:

- redesigning the user interfaces to provide information that supports the work of the flight controllers
- adding more contextual information to support the redesigned user interfaces by explicitly implementing the relationships defined in the data model
- automatically integrating data across applications, displays and processors

## 5.2 Multiple Sets of Data Stores

The mapping of data stores to the data model shown in Figure 6 is an over-simplification because there are multiple applications running on different hardware for Orbital Flight Dynamics (in

other words, the data model is implemented multiple times with little automated integration across these implementations):

- the original MOC (Mission Operations Computer), tracks 4 ephemerides for space vehicles plus 3 TDRS satellite ephemerides. MOC receives all tracking data and is responsible for updating the current ephemeris with the next state vector by using the ENCKE propagator and the Powered Flight Numerical Integrator.
- the TSA (Trajectory Sub-system Applications) is supplemental to the MOC capability. TSA can track 8 ephemerides in a global space, a further 8 ephemerides in each local space (there is a global space for Shuttle and a global space for ISS), and 8 TDRS satellite ephemerides. TSA is used for planning maneuvers, and contingency planning.
- the TSU which is a rehost of the MOC to Unix and is still under development. TSU will be able to track 10 vehicle ephemerides and 10 TDRS satellite ephemerides

Using a layered architecture approach, information would be seamlessly integrated across the MOC and TSA creating the appearance of a single data model. However, integrating information across MOC and TSA is problematic, with multiple ways of moving information around:

- a program called AutoMOC automatically mirrors the 4 vehicle ephemerides data from the MOC to TSA and the vector administration table (VAT) both ways between the MOC and TSA
- data transfer from TSA to the MOC by the flight controller operating the MEDS interface
- copy and paste from the TSA vector administration table to the MOC vector administration table, and then the flight controller must specify to which ephemeris the state vector belongs
- ephemerides cannot be transferred, they must be manually re-entered into the MOC by the flight controller (cannot get an exact duplicate because the math propagator scheme is different)

Moving data from TSA to the MOC is totally dependent on manual processing by the flight controllers. Section 5.3 reveals how much manual processing and rework is actually required.

### 5.3 Database issues

The data stores are currently implemented as flat files. The context diagram in Section 4.1 reveals how much information flows through the Orbital Flight Dynamics world, and the need to manage the flow of information.

The RIOTS project provides the opportunity to rethink systems for Orbital Flight Dynamics. In particular, should Orbital Flight Dynamics utilize database technology?

- An initial view is that the immediate priority for RIOTS should be developing a new user interface and getting the flight controllers to accept the new user interface. The underlying architecture could then be reengineered to incorporate databases as a later phase in the RIOTS project.
- More detailed appreciation for the implementation of TSA in particular has revealed that there are fundamental issues that must be addressed with an enhanced interface, including security, access control, concurrency, and file locking that Unix does not support very well and that the system developers are having to resolve in their code. In this case it makes sense to leverage database technology that has built-in these functions.
- RIOTS should employ database technology in areas where there is either:
  - new functionality that the current flat file system cannot support

- current problems in existing functionality that database technology could easily resolve

## 5.4 Constraints on Work Practice - Rendezvous Planning

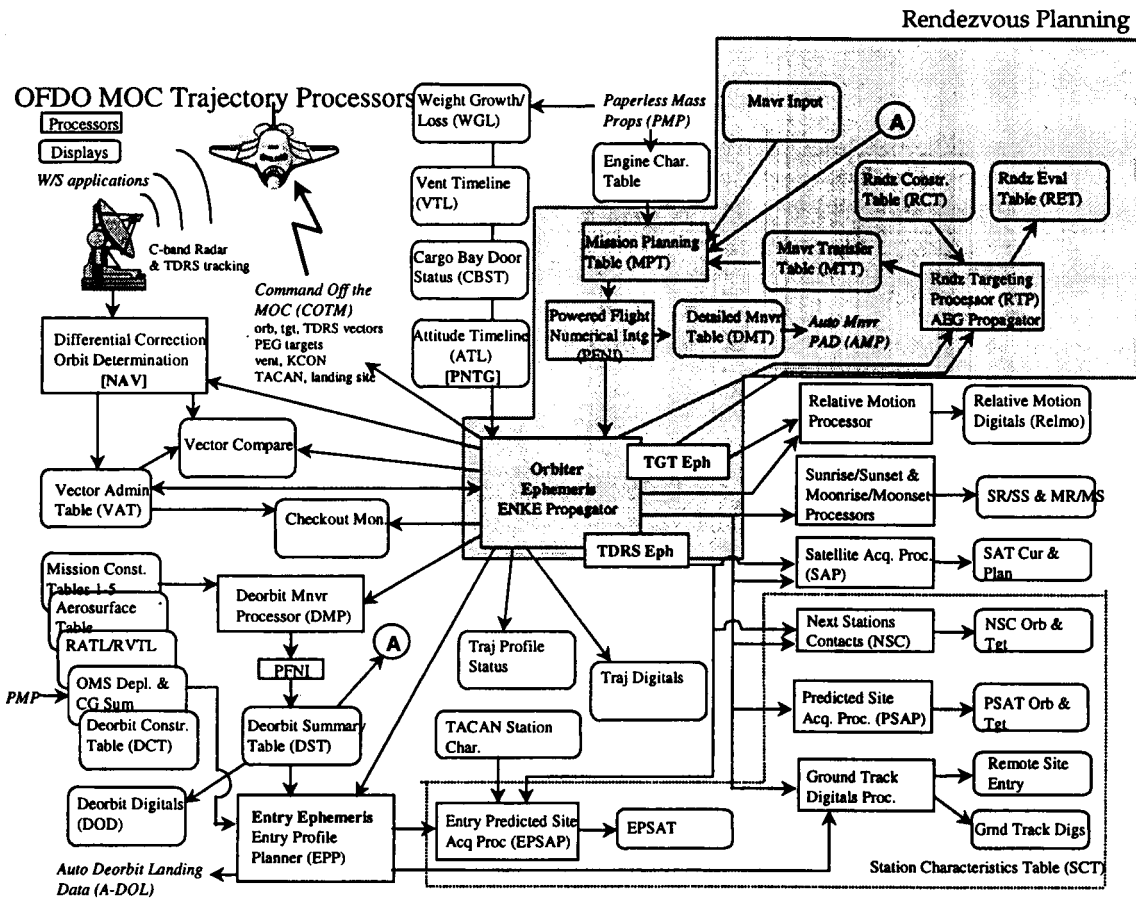


Figure 9. MOC Processors and Displays

Figure 9 was generated by members of the orbital dynamics group to illustrate the current set of MOC processors (in square-cornered boxes) and displays (in round-cornered boxes). The current system is designed around a functional input-processing-output model with CPU constraints due to outdated hardware. In this section we will examine the work required to conduct Rendezvous Planning and the constraints imposed by a characterization of work focused on hardware and software functions. The processors and displays for rendezvous maneuver planning are shown in the top right corner of Figure 9. Figure 10 shows the activity of conducting rendezvous planning.

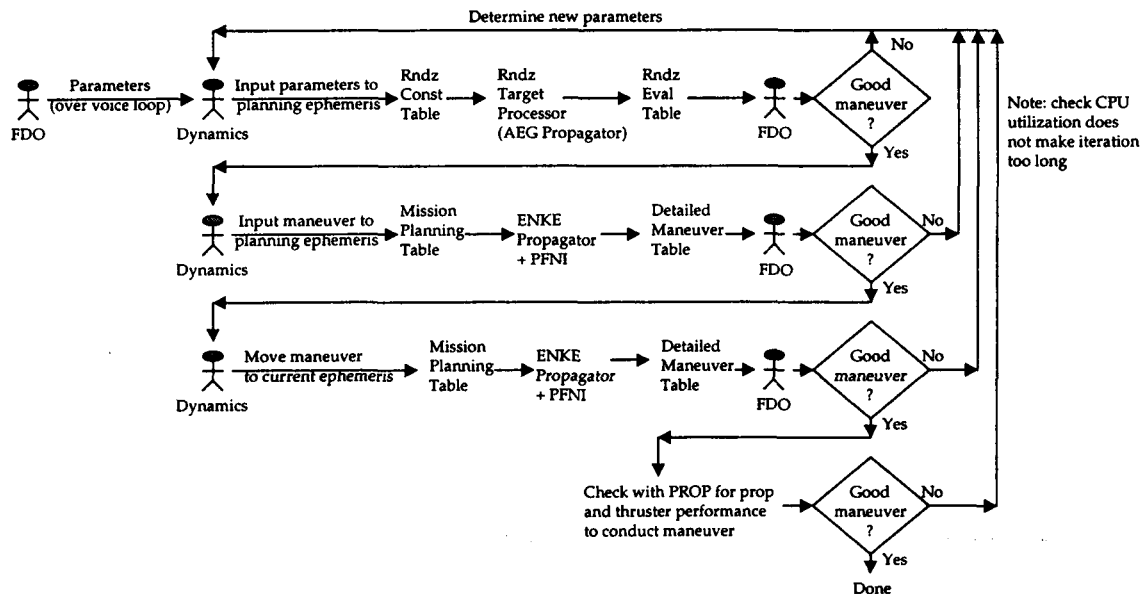


Figure 10. The activity of planning rendezvous maneuvers

Rendezvous planning is initially conducted in a planning ephemeris that uses a state vector from the current ephemeris as the anchor vector id. FDO (front-room) will tell Dynamics (back-room) over the voice loop the parameters for conducting the rendezvous. Dynamics enters the parameters into the Rendezvous Constraints Table and uses the Rendezvous Target Processor to compute a set of maneuvers to meet the specified constraints. These maneuvers can be displayed using the Rendezvous Evaluation Table. The FDO evaluates the Rendezvous Evaluation Table to determine whether the maneuver is suitable. If it is not suitable, the FDO will tell Dynamics which parameters to change and the process is repeated.

If the maneuver<sup>1</sup> is suitable, FDO tells Dynamics it is OK. Dynamics enters the maneuver into the Mission Planning Table and uses the higher fidelity ENKE propagator to update the ephemeris, creating the Detailed Maneuver Table. The FDO evaluates the Detailed Maneuver Table to check the maneuver is still suitable. If it is not suitable, the FDO will need to determine a new set of parameters to start the whole maneuver planning process.

If the maneuver is suitable, FDO tells Dynamics it is OK. Dynamics enters the maneuver in an ephemeris's Mission Planning Table and uses the ENKE propagator to update the ephemeris creating the Detailed Maneuver Table. The FDO evaluates the Detailed Maneuver Table to check the maneuver is still suitable. If it is not suitable, the FDO will need to determine a new set of parameters to start the whole maneuver planning process.

If the maneuver is suitable, FDO confirms with PROP that the propellant and thruster performance are available to execute the maneuver. If it is not suitable, the FDO will need to determine a new set of parameters to start the whole maneuver planning process. If it is suitable then the process is completed. Processing must be completed with sufficient remaining time and

<sup>1</sup> The FDOs refer to a burn as a maneuver. But the Flight Controllers seem to use the word "maneuver" to refer to changes in attitude, and the word "burn" to refer to changes in trajectory.



communication opportunity to send and read a burn PAD to the crew prior to execution.

Note: when determining a new set of parameters and restarting the maneuver planning process, a constraint on the planning activity is ensuring that the CPU utilization is not too long to complete the iteration.

Rendezvous planning is an iterative process with the potential for a lot of rework and reliance on flight controllers to manually integrate data across displays. The role relationships can be thought of as decision-making/information integration/mirroring (consider that the role of Dynamics in this situation is purely manual integration of information in response to FDO's requirements, and that the Profile/Support role is mirroring this work on a separate system). Explanations for the iterative nature of the work include:

- the first cycle of iterations is simply creating a maneuver that meets all the constraints
- the second cycle of iterations is due to the ENKE propagator being more complex than the AEG propagator by including models of engine burns, weight loss, drag and gravity. Thus a good maneuver produced by the AEG propagator may not be a good maneuver with the more complex modeling of the ENKE propagator
- the third cycle of iterations is due to divergence between the current ephemeris and the planning ephemeris. Divergence could occur because vents and maneuvers have been integrated or new tracking data has changed the anchor vector id. Rendezvous maneuver planning may occur over a long period of time (hours) with many interruptions and possibly shift changes thus making divergence between the current and planning ephemeris an issue.

#### 5.4.1 Discussion

The *iteration/rework cycle* is a recurring pattern in Orbital Flight Dynamics work practice. Deorbit planning and collision avoidance planning both rely on similar iteration/rework cycles to rendezvous maneuver planning.

The iteration/rework cycle is partly a function of the nature of Orbital Flight Dynamics work, partly a function of the input-processing-output functional design approach for each processor and display, partly due to the use of multiple propagators and ephemerides, and partly due to the lack of information integration across processors and displays.

The *decision-making/information integration/mirroring role relationship* is also a recurring pattern in Orbital Flight Dynamics work. The relationship between FDO and NAV/Track can be thought of as decision-making/information integration that combines the expertise of each discipline (and it could be argued that INCO is actually mirroring some of this work for their own purposes).

The information integration role is partly due to the lack of semantic information (see Section 5.1), the lack of integration across multiple information systems (see Section 5.2), the current work practice of relying on the human system to make things work (see Section 5.3), and due to FDO being the main point of contact for other flight disciplines (no-one calls FDO's backrooms, whereas other flight disciplines regularly have their backrooms called).

## 6 Envisioning the Future Work of Orbital Flight Dynamics

New requirements are changing the shape of Orbital Flight Dynamics work. The first new requirement is the need to be able to share a global current Orbiter ephemeris to the whole flight control team at all times. The second new requirement is the ongoing capability for conducting missions to support Orbiter flights to ISS and the need to integrate information across mission control centers (at least Shuttle MCC and ISS MCC at JSC and possibly others). The third new requirement is the intent to utilize the extra ephemerides provided by the TSA and TSU software.

The fourth requirement is to increase the level of interaction between FDO and the flight control team so that the probability of mission success remains high while the number and magnitude of constraints on trajectory operations increase.

Implementing these new requirements utilizing the current work patterns of the iteration/rework cycle and the decision-making/information integration/mirroring roles represents a significant increase in the workload for Orbital Flight Dynamics and will require an increase in manpower. Increasing manpower is problematic for many reasons, including staffing, training and an increasingly demanding flight schedule. We must find a way to reduce the cognitive workload of the flight controllers and redistribute the work within the Orbital Flight Dynamics group.

Reducing the workload requires changing the current work patterns. In particular:

- by automating the information integration role. Automating the information integration role includes:
  1. Ideally rearchitecting TSA and TSU to use a layered architecture fully implementing the data model by using database servers. However, the immediate priority for TSU is rehosting the MOC functionality.
  2. Implementing semantics for ephemerides and all planning objects by instantiating the relationships defined in the data model in Section 4. The semantics include:
    - a field defining the purpose of the ephemeris, including it's assumptions and contingencies.
    - a field defining the heritage of the ephemeris
    - a field defining the currency of the ephemeris (time elapsed since last change / update)
    - a field tracking changes to the ephemeris (for example, new, update, delete for maneuver, vent, weight, anchor vector id, length)
    - a field defining the purpose of the maneuver, vent, weight change
    - a field defining the ephemeris using the object for the maneuver, vent, state vector, weight and attitude (there may be multiple ephemerides for each object)
    - a field linking weight loss to a maneuver or vent (each maneuver and vent should automatically generate an associated weight change, similarly with deploys and retrievals)
    - a field defining the status of the maneuver planning (is it good for a certain time, finished)
    - for each maneuver, the system should automatically generate a parameter trials object that tracks: each set of parameters, and ideally the results of each parameter trial and reasons for change
  3. Utilize the semantics data for ephemerides to enable comparison between current and planning ephemerides before inserting a maneuver into the current ephemeris. FDO should be informed of differences, and have the opportunity to automatically update the planning ephemeris, rerun the parameter trial, and inform the flight controller of changes to the ephemeris and the results of the new parameter trial. Alternatively, a history of postponed updates should be maintained so that FDO can implement selected updates as needed.
  4. Create agents that automatically generate MEDS commands, removing the need to reenter data to move a good maneuver to the next propagator / ephemeris (see Figure 9).

- by reducing the number of iterations required in the iteration/rework cycle:
  1. For all planning situations that use two propagators, analyze why multiple propagators are being used:
    - Is it because of computational limitations (time to produce output)? If so, identify the limitation and solve by getting a faster CPU, more memory, or reimplementing the propagator.
    - Is it because of resource limitations (too many things try to use the propagator)? If so, consider putting copies of the propagator on other machines, increasing the amount of available resource.
    - Is it due to some issue in modeling maneuvers? If so, identify the issue and seek to improve the existing models.
    - If multiple propagators must be used, can we identify some way of perturbing the inputs to the first propagator to produce the outputs required for the second?
  2. Create agents that assist tracking of the most accurate state vectors for maneuvers and rendezvous. For example, knowing that the onboard state vector is most accurate during rendezvous and after maneuvers (for the first orbit). The agents should be able to detect the drift of accuracy of the ephemeris state vector and alert the flight controller when the drift is nearing limits. At all times, the agents should provide the flight controllers with visibility of the underlying data.
  3. Implementing heuristics for decreasing iterations with the aim of sharing detailed expert planning knowledge across the Orbital Flight Dynamics group:
    - identify heuristics from parameter trials by expert FDOs with the aim of reducing the number of iterations
    - after each mission/sim, automatically generate an analysis of maneuver planning as a tool for aiding improvement and learning in the Orbital Flight Dynamics group
- by creating an enterprise data model for Mission Control (initially Shuttle and then expand to ISS). One way of viewing an enterprise data model is that it takes the context diagram (see Section 4.1) for each flight control discipline, converts the information flows across flight disciplines into a data model, and the enterprise data model is then the union of all these cross-discipline data models. The enterprise data model focuses on who/what creates, updates, deletes, and touches the data.
- the enterprise data model provides the basis for tracking and communicating changes to an ephemeris:
  1. Utilize the semantic data to track changes to an ephemeris. All users of the ephemeris (see Figure 4) should be able to register an interest in the ephemeris using a publish/subscribe type mechanism. Whenever the current ephemeris is changed, a message should be sent to all registered agents/flight controllers stating that there is change, what the change is, and when the change will be implemented. Agents and flight controllers can use these messages to determine when they need to update their components.
  2. Enable Orbital Flight Dynamics customers to know when the underlying data has changed. For example the AOS Acquisition Monitor for the GCs and INCOs is not a dynamic display and provides no way of knowing whether changes to the ephemeris require the AOS Acquisition times to be recalculated.

Redistributing the work within the Orbital Flight Dynamics group requires rethinking the decision-making/information integrating/mirroring role pattern. Can we reconceive the roles in terms of:

- FDO performing short-term proactive work with the flight control team and managing the integration of the detailed planning products
- NAV, Dynamics and Profile Support conducting detailed planning with secondary duties of monitoring tracking data divergence (NAV), knowledge about the internal workings of the information systems (Dynamics), and mirroring planning of critical maneuvers (Profile Support)

The aim of the redistribution is to reduce the need for manual integration, increase the number of people available to conduct the more cognitively demanding detailed planning utilizing the availability of more ephemerides, and free FDO to interact more with the Flight Control Team

We can provide the infrastructure for supporting these new role patterns by reducing the information integration requirements, simplifying the iteration/rework cycle and implementing learning processes that share detailed planning knowledge across the Orbital Flight Dynamics group.

## 7 Conclusions

The collaborative, iterative development of the preliminary data model and context diagram with the Orbital Flight Dynamics group surfaced knowledge about the nature of work, the underlying information requirements, and the underlying information systems issues that were making development and maintenance problematic.

Given the historical emphasis on functional analyses in Mission Control, we recommend conducting an information-centric analysis in each flight discipline for both Shuttle and ISS. An outcome of these information-centric analyses is an enterprise data model that is owned by MOD. The enterprise data model would be used to identify information disconnects, redundancy and rework with the aim of redesigning the Mission Control work system to manage the creation and flow of information, improving performance by reducing errors and ensuring the consistency of information across the entire Flight Control Team.

An information-centric analysis of Orbital Flight Dynamics reveals many information disconnects that are currently managed by wasteful reliance on human labor. The information disconnects can be traced to partial implementation of the data model, having multiple information systems implement the data model and relying on manual integration of information across information systems and work practices.

Fixing the information disconnects requires a combination of information systems initiatives to reduce the information integration requirements on the flight controllers and work system design initiatives to redesign the distribution of work within Orbital Flight Dynamics. We would recommend investigating database technology for the RIOTS project primarily to overcome the Unix flat file deficiencies described in Section 5.3, and secondly as an information store.

The context diagram in Section 4.1 reveals the magnitude of interdependencies between Orbital Flight Dynamics and other flight disciplines. The next step in work systems redesign is to make these interdependencies visible across the entire flight control team to support the Flight Director's decision-making, and then develop new practices to ensure that these interdependencies do not collapse in problem situations.

## 8 References

Chen, P.P.S. (1976).

The classic reference for data modeling is: "The Entity-Relationship Model -- toward a unified view of data". *ACM Transactions on Database Systems* 1, 1, March 1976.

There are many books, web sites, and courses on data modeling. For information about the different types of models in a data modeling approach: [http://www.jayzed.com/data\\_model.htm](http://www.jayzed.com/data_model.htm).

For information about the contents of a data model: <http://www.islandnet.com/~tmc/html/articles/datamodl.htm>.

An example of the type of topics that should be covered in a data modeling course see: [http://www.infoimpact.com/body/course\\_cdm.html](http://www.infoimpact.com/body/course_cdm.html).

RIOTS (2000). "Redesigned Interface for Operational Trajectory Support (RIOTS) Part 1: Ephemeris Maintenance LEVEL A SOFTWARE REQUIREMENTS", Version 1, April 18, 2000 Mission Operations Directorate, Flight Design and Dynamics Division.

Shalin, V.L. and Geddes, N.D. (1994). Task dependent information management in a dynamic environment: Concept and measurement issues. Invited Session on Associate Systems. *Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics*, (pp. 2102-2107). San Antonio, TX, October.

Shalin, V.L. and Geddes, N.D. (1998). Pilot performance with mid-flight plan-based display changes. *Proceedings of HICS '98: 4th Annual Symposium on Human Interaction with Complex Systems* (pp. 170-181). Dayton, OH.